In [2]:
```
import os
import pandas as pd
import torch
from PIL import Image
from torch.utils.data import Dataset

class RetinaDataset(Dataset):
    def __init__(self, csv_file, root_dir, transform=None, extension='.jpeg'):
        print("Initializing RetinaDataset with extension support")  # Debug print
        """
        Args:
            csv_file (string): Path to the csv file with annotations.
            root_dir (string): Directory with all the images.
            transform (callable, optional): Optional transform to be applied on a
            extension (string, optional): File extension for all images.
        """
        self.labels_frame = pd.read_csv(csv_file)
        self.root_dir = root_dir
        self.transform = transform
        self.extension = extension

    def __len__(self):
        return len(self.labels_frame)

    def __getitem__(self, idx):
        img_name = os.path.join(self.root_dir, self.labels_frame.iloc[idx, 0] + s
        image = Image.open(img_name)
        label = self.labels_frame.iloc[idx, 1]

        if self.transform:
            image = self.transform(image)

        label = float(self.labels_frame.iloc[idx, 1])
        label = torch.tensor(label, dtype=torch.long)
        return image, label
```

In [ ]:
```
```